Journal of the Operations Research
Society of Japan
Vol. 24, No. 1, March 1981

© 1981 The Operations Research Society of Japan

WEIGHTED MINIMAX REAL-VALUED FLOWS

Tetsuo Ichimori
Osaka University

Hiroaki Ishii
Osaka University

Toshio Nishida
Osaka University

(Received May 6, 1980; Final September 3, 1980)

Abstract An algorithm for solving a weighted minimax real-valued flow problem in a polynomial time is given. The weighted minimax flow is that which minimizes the maximum value of arc-flow multiplied by arc-weight among all flows of maximum flow value. We use the capacity modification technique and the method of solving a ratio minimization problem for our problem.

1. Introduction

A *minimax flow* and a *weighted minimax flow* have been discussed in [2,3,5]. A (weighted) minimax flow is that which minimizes the maximum value of arc-flow (multiplied by arc-weight) among all flows of maximum flow value. A minimax flow (or an unweighted minimax flow) problem is solvable within $O(m_0 n^3)$ time where n is the number of nodes in a network and m_0 is the number of arcs across the minimum cut in the network [2]. A weighted minimax flow problem is also solvable within a polynomial time where capacities, weights and arc flows are all required to have nonnegative integer values [5]. If they are non-negative real numbers, the weighted minimax flow problem can not be solved within a polynomial time by the algorithm in [5]. We note that the minimax flow problem can be solved in either an integer form or a real form. Hence we wish to obtain a weighted minimax real-valued flow in a polynomial time.

For the applications of (weighted or unweighted) minimax (integer or real) flows we have the *sharing problem* [4] and the *storage management problem* [9].

2. Weighted Minimax Real-Valued Flow

Let $G=(N,A)$ be a network where N is the set of nodes and A is the set of directed arcs connecting nodes. A source s and a sink t are designated. With each arc (i,j) in the set A we associate a real positive capacity $c(i,j)$ and a nonnegative real weight $w(i,j)$. A *flow* is denoted by f. Given a flow f, we refer to $f(i,j)$ as the arc flow $f(i,j)$. We assume for simplicity that source s has no arcs incident into s and that sink t has no arcs incident out of t.

The weighted minimax real-valued flow problem is:

(1) $$\min[\max_{(i,j)\in A} w(i,j)f(i,j)]$$

s.t.

(2) $$\textstyle\sum_{(i,j)\in A} f(i,j) = \sum_{(j,i)\in A} f(j,i), \quad j\neq s,t$$

(3) $$\textstyle\sum_{(s,i)\in A} f(s,i) = v^*$$

(4) $$\textstyle\sum_{(i,t)\in A} f(i,t) = v^*$$

(5) $$0 \le f(i,j) \le c(i,j), \quad (i,j)\in A$$

where v^* is the value of the maximum flow from source s to sink t.

For the above problem we use the *capacity modification technique* in [2] and the strategy for solving *ratio minimization problem* in [8]. For a nonnegative parameter D, we define the new capacity $c'(i,j)$ of each arc (i,j) as follows:

(6) $$c'(i,j) = \min(D/w(i,j),\ c(i,j)), \quad (i,j)\in A$$

where if $w(i,j)=0$, we define $c'(i,j)=c(i,j)$. The capacity of a *cut* (S,T) is denoted by $c(S,T)$ where S denotes a subset of N such that $s\in S$ and T denotes the complement of S in N such that $t\in T$. The value of $c(S,T)$ is defined as follows:

$$c(S,T) = \textstyle\sum_{(i,j)\in(S,T)} c(i,j).$$

We define the new capacity of the cut (S,T)

$$c'(S,T) = \textstyle\sum_{(i,j)\in(S,T)} c'(i,j).$$

By definition we have $c(S,T)\ge c'(S,T)$ for any cut (S,T). By the famous *max-flow min-cut theorem* [1] we have $v^*=\min c(S,T)$. Let

$$v(D) = \min_{(S,T)\text{ in all cuts}} c'(S,T).$$

Note that $v(D)$ is nondecreasing and piecewise linear function of D. Let D^* be the minimum value of D such that $v(D)=v^*$. Then we have $D^*\ge f(i,j)w(i,j)$ for any arc $(i,j)\in A$, where f is a feasible flow. Therefore it is obvious that D^* is the optimal value of (1).

Next consider the problem of finding D^*. Given a network G, consider the new network, denoted by $G(D)$, with capacities $c'(i,j)$ instead of $c(i,j)$. Note

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

that if D is infinite, then $G(D)$ and G are identical and that $v(D)$ is the value of the maximum flow in $G(D)$. Then, if $v(D)$ is strictly less than v^*, D is smaller than D^*. Otherwise, D is larger than or equal to D^*. This is very similar to the case of solving ratio minimization problems [7]. Now we develop an algorithm for our problem based on a similar idea to Megiddo's [8].

We define $d(i,j)=c(i,j)w(i,j)$ for each arc (i,j), set D' to the maximum value of $d(i,j)$ such that $v(d(i,j))< v^*$, and set D'' to the minimum value of $d(i,j)$ such that $v(d(i,j))=v^*$. Then we have $D'<D^*\le D''$.

The computation of a maximum flow involves operations (+, -, min) between two data. In general, data are integers or reals. However we consider the case where data are of linear form $aD+b$ (a and b are reals). Then for two data, $aD+b$ and $a'D+b'$ where $a\ge a'$, we have the following:

$$(aD+b) + (a'D+b') = (a+a')D+(b+b')$$

$$(aD+b) - (a'D+b') = (a-a')D+(b-b')$$

$$\min(aD+b,\ a'D+b') = \begin{cases} aD+b & \text{for } D>B \\ a'D+b' & \text{for } D\le B, \end{cases}$$

where B is an intersection point of $aD+b$ and $a'D+b'$, i.e.

$$B = (b'-b)/(a-a'),\ a\ne a'.$$

If D is restricted so that $D>B$ or $D\le B$, then these operations (+, -, min) between two data, $aD+b$ and $a'D+b'$ can be treated in a similar manner as for integers or reals.

Since data $aD+b$ involve parameter D we call them parametric. When "parametrically" is referred to in this paper, we intend "by the operations between parametric data."

Next we seek a maximum flow parametrically in $G(D)$, $D'<D\le D''$. Note that at the start of this computation, data are $D/w(i,j)$, $D'<D\le D''$ or $c(i,j)$. During the computation, the min operations may demand the restriction of D, i.e., $D'<D\le B$ or $B<D\le D''$, in order to determine $\min(aD+b,\ a'D+b') = aD+b$ or $a'D+b'$. In this case we put pause to the parametric computation of the maximum flow in $G(D)$, $D'<D\le D''$. Instead of it we compute $v(B)$. Note that in this computation all data are constant but not parametric. If $v(B)< v^*$, then we have $B<D^*\le D''$; else $D'<D^*\le B$. Hence we replace D' or D'' by B according to the value of $v(B)$ and resume the preceding parametric computation. At the end of this process we get the value of the maximum flow in $G(D)$, which is of form a^*D+b^*. If $a^*\ne 0$, then $D^*=(v^*-b^*)/a^*$; otherwise $D^*=D''$.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Algorithm

Step 1. Set $D' = \max\{d(i,j) \mid v(d(i,j)) < v^*\}$.
Set $D'' = \min\{d(i,j) \mid v(d(i,j)) = v^*\}$.

Step 2. Compute the maximum flow parametrically in $G(D)$, $D'<D\leq D''$, from the start or from the recent point of resumption, to the next point of restrictions of D. If there are no more restrictions of D, then we have the value of the maximum flow (denoted by a^*D+b^*) and go to Step 4; otherwise let the parametric computation of the maximum flow make a pause at the point of restriction of D and go to Step 3.

Step 3. Let B denote an intersection point involved by the current restriction of D where $D'<B\leq D''$. Compute $v(B)$. If $v(B)< v^*$, then set D' to B; otherwise set D'' to B, and go back to Step 2 to resume the parametric computation of the maximum flow.

Step 4. If $a^*\neq 0$, compute $D^*=(v^*-b^*)/a^*$; otherwise $D^*=D''$. Stop.

Remark. The optimal flow which gives D^* is determined by the maximum flow in $G(D^*)$.

Illustration

Consider the graph $G=(N,A)$ depicted in Fig. 1, where the first number of the label on each arc is its capacity and the second number is its weight.

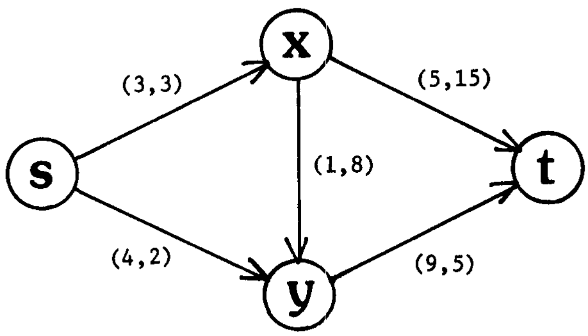

Fig. 1. $G=(N,A)$. The label $(c(i,j),\ w(i,j))$ denotes (capacity, weight) of arc (i,j).

The value of the maximum flow in $G=(N,A)$, v^* is 7. A description of the algorithm is given below.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Step 1. Arc (s,x) gives D'=9 and arc (y,t) gives D'=45 since $v(9)$ = 12/5 < 7 (see Fig. 2) and $v(45)$ = 7 (see Fig. 3), respectively.

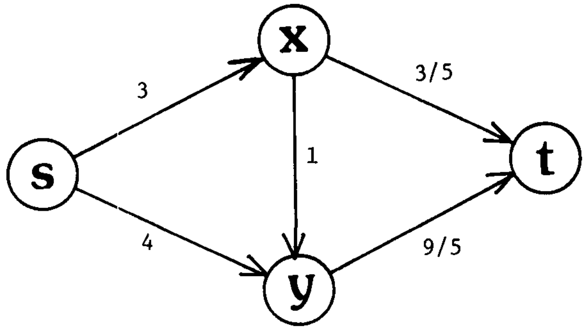

Fig. 2. Graph $G(9)$. The number on each arc is its capacity, $c'(i,j)$ which is given by the following equation: $c'(i,j) = \min(9/w(i,j),\ c(i,j))$. Arcs (x,t) and (y,t) determine the value of $v(9)$ as: $v(9) = 3/5 + 9/5 = 12/5$.

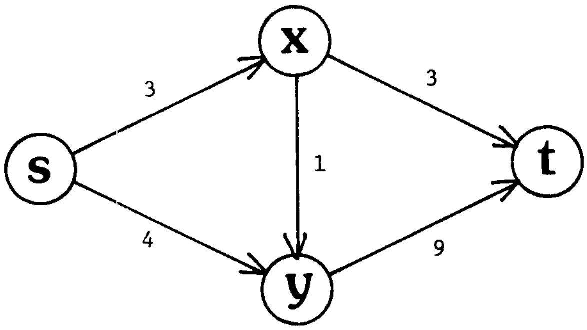

Fig. 3. Graph $G(45)$. Each capacity: $c'(i,j) = \min(45/w(i,j),\ c(i,j))$. Arcs (s,x) and (s,y) give the value of $v(45)$ as: $v(45) = 3 + 4 = 7$.

Step 2. We compute the maximum flow parametrically in $G(D)$, $9<D\le 45$, (see Fig. 4). We choose the path s-y-t and send the largest possible flow through this path. This involves the computation min(4, D/5). Since

$$\min(4,\ D/5) = \begin{cases} D/5, & 9<D\le 20 \\ 4, & 20<D\le 45, \end{cases}$$

the restriction of D is demanded. We put pause to this computation with respect to $G(D)$ and go to Step 3 for determining whether min(4, D/5)= D/5 or 4.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

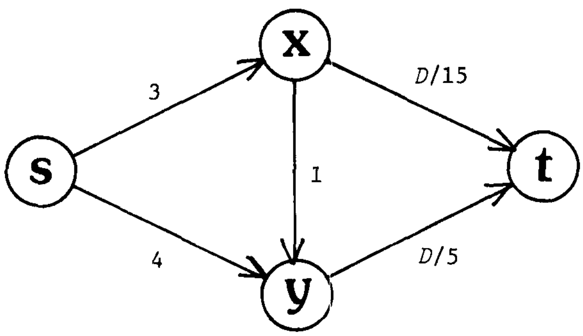

Fig. 4. $G(D)$, where the number on each arc is its capacity: $c'(i,j) = \min(D/w(i,j),\ c(i,j))$, $9<D\leq 45$.

Step 3. We get $B=20$ and $v(20)=16/3$. See Fig. 5 where $c'(i,j) = \min(20/w(i,j),\ c(i,j))$.

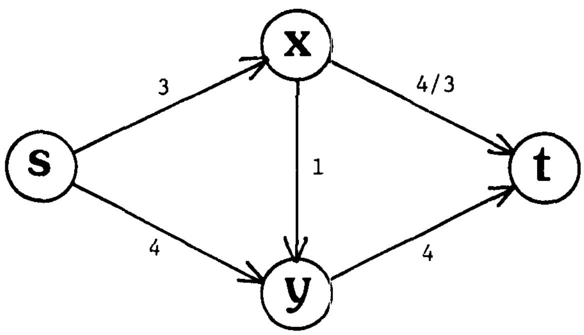

Fig. 5. Graph $G(20)$. $v(20) = 4/3 + 4 = 16/3$ since arcs (x,t) and (y,t) are saturated.

Since $v(20) < 7$, we have $D'=20$. Note $20<D^*\leq 45$.

Step 2. Step 3 above yields $\min(4,\ D/5) = 4$, $20<D\leq 45$. Let $f(i,j)$ denote the arc flow on arc (i,j). Then $f(s,y)=f(y,t)=4$. Next we choose the path s-x-y-t from $G(D)$. Then the following computation is necessary: $\min(3,\ 1,\ D/5 - 4)$. Since

$$\min(1,\ D/5 - 4) = \begin{cases} D/5 - 4, & 20<D\leq 25 \\ 1, & 25<D\leq 45, \end{cases}$$

again we put pause to this computation and go to Step 3.

Step 3. Since $B=25$ and $v(25)=20/3 < 7$, we set $D'=25$. Note $25<D^*\leq 45$.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

Step 2. Since we get min(3, 1, $D/5 - 1$) = 1, $25<D\le 45$, we have $f(s,x)=f(x,y)=1$ and $f(y,t)=4+1=5$. Next we choose the path s-x-t from $G(D)$, which needs the computation min(3 - 1, $D/15$). Here

$$\min(2,\ D/15)=\begin{cases} D/15, & 25<D\le 30 \\ 2, & 30<D\le 45. \end{cases}$$

Step 3. Since $B=30$ and $v(30)=7$, we set $D''=30$. Note $25\lneq D^*\le 30$.

Step 2. Since min(2, $D/15$) = $D/15$, $25<D\le 30$, we have $f(s,x)=1+D/15$, $f(x,t)=D/15$. Now the maximum flow is obtained since there is no path capable of sending a flow, (see Fig. 6).

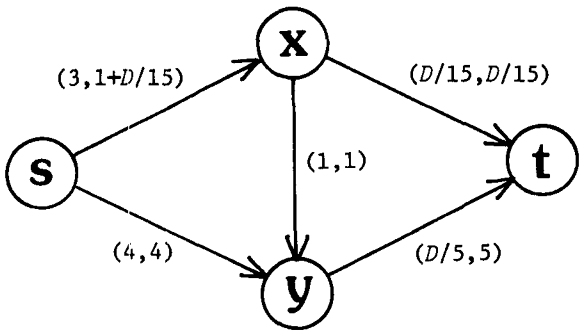

Fig. 6. Graph $G(D)$. First label is arc capacity and second label is arc flow. Arcs (s,y), (x,y) and (x,t) are saturated.

The maximum flow value, a^*D+b^* is $D/15 + 5$.

Step 4. $D^*=15(7-5)=30$, (see Fig. 7).

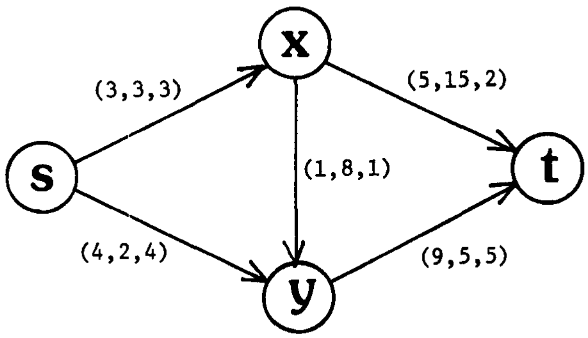

Fig. 7. Optimal flow with $D^*=30$. The label on each arc is (capacity, weight, flow).

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

3. Time Complexity

It takes $O(n^3 \log n)$ time to obtain D' and D'' in Step 1 by a binary search [7,8], which is not so important here since the time bound is negligible as compared with that in Step 3. The maximum flow is obtained in time $O(n^3)$ [6]. Therefore Step 2 takes $O(n^3)$ time though it has many interruptions. (Note that the computation of finding the maximum flow in $G(D)$ parametrically is done exactly once.) Since it is obvious that the number of interruptions in Step 2 is bounded by the number of comparisons of parametric data and since all of the comparisons must be included in the $O(n^3)$ operations for finding the maximum flow in Step 2, Step 3 is iterated at most $O(n^3)$ times, where each iteration needs the computation of $v(B)$ whose time bound is $O(n^3)$. Hence Step 3 needs a total of $O(n^6)$ time. Therefore the overall time complexity of the algorithm is $O(n^6)$. We obtain the desired result.

References

[1] Ford, L. R., Jr., and Fulkerson, D. R.: *Flows in Networks*. Princeton University Press, Princeton, 1962.

[2] Ichimori, T., Ishii, H., and Nishida, T.: A Minimax Flow Problem. *Mathematica Japonica*, 24 (1979), 65-71.

[3] Ichimori, T., Murata, M., Ishii, H., and Nishida, T.: Minimax Cost Flow Problem. *Technology Reports of the Osaka University*, 30 (1980), 39-44.

[4] Ichimori, T., Ishii, H., and Nishida, T.: Optimal Sharing. (Submitted).

[5] Ichimori, T., Ishii, H., and Nishida, T.: Finding the weighted Minimax Flow in a Polynomial Time. *Journal of the Operations Research of Japan*, 23 (1980), 268-272.

[6] Karzanov, A. V.: Determining the Maximal Flow in a Network by the Method of Preflow. *Soviet Mathematics Doklady*, 15 (1974), 434-437.

[7] Lawler, E. L.: *Combinatorial Optimization; Networks and Matroids*. Holt, Rinehart and Winston, New York, 1976.

[8] Megiddo, N.: Combinatorial Optimization with Rational Objective Functions. *Mathematics of Operations Research*, 4 (1979), 414-424.

[9] Stanat, D. F. and Magó, G. A.: Minimizing Maximum Flows in Linear Graphs. *Networks*, 9 (1979), 333-361.

Tetsuo ICHIMORI: Department of Applied Physics, Faculty of Engineering, Osaka University, Yamada-Kami, Suita, Osaka 565, Japan

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.